

A PROPOSAL FOR ONLINE COURSE AT THE ENGINEERING FACULTY

Adrian Runceanu, “Constantin Brancusi” University, Targu-Jiu, ROMANIA
Marian-Ion Popescu, “Constantin Brancusi” University, Targu-Jiu, ROMANIA

Abstract: *In this article we try to present an implementation of the undergraduate online Computer Programming course that we have implemented in the Moodle platform at the Engineering Faculty. This online course has been developed as an enhancement of the face-to-face course. The aim of this course is to create an online learning community which helps both teachers and students to have a virtual space where we could use a tool for practical activities in online environment. This is just a first course that can be used in engineering study field. We believe that all the other study programs (Control Systems, Electronics, Robotics, Electrical engineering, Mechanics, Environment Protection in Industry, and others) could use this idea for develop specific courses.*

Key Words: *Virtual Laboratory, Practical Laboratory Activities, Moodle, E-learning, Computer Programming, Collaborative learning, Social Constructionist Pedagogy.*

1. INTRODUCTION IN MOODLE

Moodle is a Course Management System (CMS), also known as a Learning Management System (LMS) or a Virtual Learning Environment (VLE). It is a Free web application that educators can use to create effective online learning sites.

Moodle is a free learning management system that enables you to create powerful, flexible, and engaging online learning experiences.

The word Moodle was originally an acronym for Modular Object-Oriented Dynamic Learning Environment, which is mostly useful to programmers and education theorists. It's also a verb that describes the process of lazily meandering through something, doing things as it occurs to you to do them, an enjoyable tinkering that often leads to insight and creativity.

Moodle is designed to support a style of learning called Social Constructionist Pedagogy. This style of learning is interactive. The social constructionist philosopher believes that people learn best

when they interact with the learning material, construct new material for others, and interact with other students about the material.

Moodle enables you to add five kinds of static course material. This is course material that a student reads, but does not interact with:

- A text page
- A web page
- A link to anything on the Web (including material on your Moodle site)
- A view into one of the course's directories
- A label that displays any text or image

However, Moodle also enables you to add six types of interactive course material.

This is course material that a student interacts with, by answering questions, entering text, or uploading files:

- 1) Assignment (uploading files to be reviewed by the teacher and/or students)
- 2) Choice (a single question)
- 3) Journal (an online journal)
- 4) Lesson (a conditional, branching activity)
- 5) Quiz (an online test)
- 6) Survey (with results available to the teacher and/or students)

Moodle also offers five kinds of activities where students interact with each other.

These are used to create social course material:

- 1) Chat (live online chat between students)
- 2) Forum (you can choose the number of online bulletin boards for each course)
- 3) Glossary (students and/or teachers can contribute terms to site-wide glossaries)
- 4) Wiki (Wikis can be inserted into courses, or a Wiki can be the entire course)
- 5) Workshop (workshops support collaborative, graded efforts among students)

2. ADVANTAGES OF MOODLE

First of all, Moodle is an open source, is built on a sound educational philosophy, and has a huge community that supports and develops it. It can compete with the big commercial systems in terms of feature sets and is easy to extend.

2.1. Free and Open Source

The idea itself is simple: open source simply means that users have access to the source code of the software. You can look under the hood, see how the software works, tinker with it, share it with others, or use parts of it in your own product.

So why is this important? For one, open source software is aligned with the academic community's values of freedom, peer review, and knowledge sharing. Just as anyone can download and use Moodle for free, users can write new features, fix bugs, improve performance, or simply learn by seeing how other people solved a programming problem. Secondly, unlike expensive proprietary CMSs that require license fees and maintenance contracts, Moodle costs nothing to download and you can install it on as many servers as you want. No one can take it away from you, increase the license cost, or make you pay for upgrades. No one can force you to upgrade, adopt features you don't want, or tell you how many users you can have.

2.2. Educational Philosophy

This is revolutionary, as most CMS systems have been built around tool sets, not pedagogy. Most commercial CMS systems are tool-centered, whereas Moodle is learning-centered. Social constructionism is based on the idea that people learn best when they are engaged in a social process of constructing knowledge through the act of constructing an artifact for others.

While toolcentric CMSs give you a list of tools as the interface, Moodle builds the tools into an interface that makes the learning task central. You can organize your Moodle course by week, topic, or social arrangement. Additionally, while other CMSs support a content model that encourages instructors to upload a lot of static content, Moodle focuses on tools for discussion and sharing artifacts. The focus isn't on delivering information; it's on sharing ideas and engaging in the construction of knowledge. Moodle's design philosophy makes this a uniquely teacherfriendly package that represents the first generation of educational tools that are truly useful.

2.3. Community

Moodle has a very large, active community of people who are using the system and developing new features and enhancements. You can access this community at <http://www.moodle.org> and enroll in the Using Moodle course. There you'll find people who are more than willing to help new users get up and running, troubleshoot, and use Moodle effectively.

As of this writing, there are over 400,000 people registered on Moodle.org and over 55,166 Moodle sites in 231 countries.

The global community has also translated Moodle into over 70 languages.

These three advantages - open source, educational philosophy, and community - make Moodle unique in the CMS space.

The wide community of "Moodle" yet existing is a great advantage over every other platform. Both, multilingual web forums and comprehensive documentation are very

helpful on problem-solving of most different kinds.

3. VIRTUAL PROGRAMMING LAB – A SHORT PRESENTATION

VPL [3] - Virtual Programming Lab is a activity module that manage programming assignments and whose salient features are:

- Enable to edit the programs source code in the browser
- Students can run interactively programs in the browser
- You can run tests to review the programs.
- Allows searching for similarity between files.
- Allows setting editing restrictions and avoiding external text pasting.

3.1. General features [1]

VPL is composed of three elements: a Moodle's module, a browser-based code editor and a jail component (Fig. 1).

The code editor is a java applet providing basic features to edit, run, debug and evaluate programs code in a very simple code-development environment. To use full features a web browser with JavaScript and support for Java 1.5 applets is required.

The Moodle's module provides the typical features of this kind of component (backup and restore, integration with the grading book,

course reset, events control, role-based access) but also specific features such as: submission management, assessment support and anti-plagiarism features. The VPL module requires a Moodle version 1.9.x and PHP5 or higher. The jail component is the server in charge of compile and executes the code submitted by the students in a secure environment.

3.2. Types of VPL activities

VPL can be used to configure, manage and assess a range of learning activities which can be classified by type or scope [2]. By type, the learning activities may be: examples, completing or puzzle exercises, and code development exercises. By scope they may be: out-classroom tasks, or in-classroom exams.

3.2.1. Learning activities by type

Examples are activities where the students are provided with both the description of a problem and the program code that solves that problem. The students may interact with the code (running or debugging) to see how it works.

Examples may be mutable or immutable, depending on if the student can or no modify the code. An immutable example must be marked as "example" in the activity's configuration window. The student who modifies the code of a mutable example can always reset it to its original state.

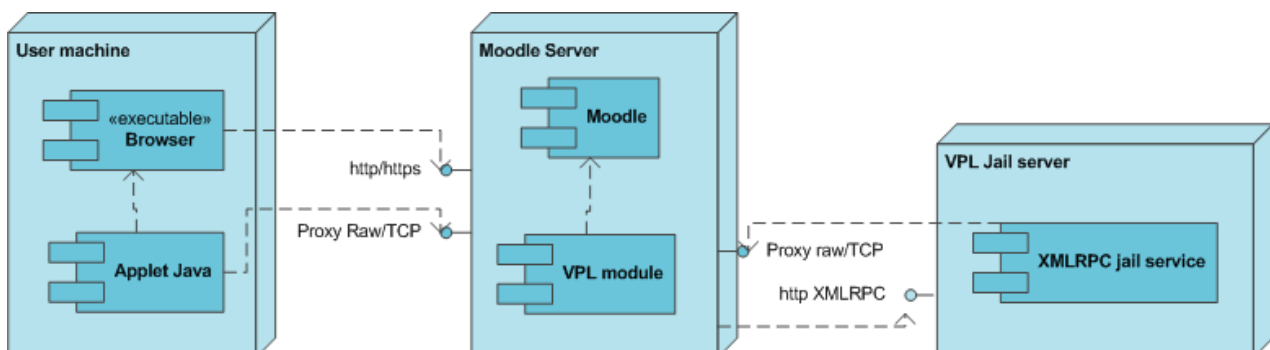


Figure 1. VPL components

Completing exercises and puzzle exercises are especially appropriate for beginners. In the same way than the previous activity, they provide the students with the description of a problem and the program code that solves that

problem, but now the student must modify the code.

In the case of the completing exercises, portions of the code have been deleted and the student has to fill in the blanks in order to do the code works in the appropriate way.

In the case of the puzzle exercises, the code has been disordered and the student has to sort it in order to do the code works in the appropriate way.

A code development exercise provides the students only with the description of a problem (although some code may be sometimes included). The student has to

3.2.2. Learning activities by scope

Attending to the conditions under what a VPL's activity can be done; there are two ways to use VPL: out-classroom tasks, or in-classroom exams.

Out-classroom tasks are designed as long-term activities to be done into a time period that may extend by days or weeks. During the active period of the activity the student may try so many solutions as he or she likes.

Out-classroom activities can be done anywhere, without direct supervision of an instructor, although help can be obtained from the instructors as long as they offer it in some way (e-mail, forums, face-to-face).

In-classroom exams are activities designed to be done in a short-term and in a restricted environment, under instructor's supervision. So, they can be configured to require a password, be done in a specific local area network, or do not permit the editor's copy and paste features.

develop the code to solve the problem using the appropriate techniques. This kind of exercises is a traditional way to achieve programming skills at any level by means of an intensive training in solving problems, in the believe that: 'doing

As for our-classroom activities, multiple attempts of solution could be tried. In both cases, the student can get immediate feedback for each try, if so configured.

4. AN IMPLEMENTATION FOR ENGINEERING FACULTY

We have developed a website for the Faculty of Engineering courses, through which students to present information about different programming courses taught at study. On the other hand we want to improve student assessment by teachers and also to offer a modern self-evaluation of students. (Fig. 2) Access to the site is www.scoaladeinformatica.ro clicking moodle and then entering identification data received from the teacher who manages the course / courses on site. Each student can select the course you want to log by selecting the name of the course. (Fig.3)



Figure 2. Engineering Faculty login first page

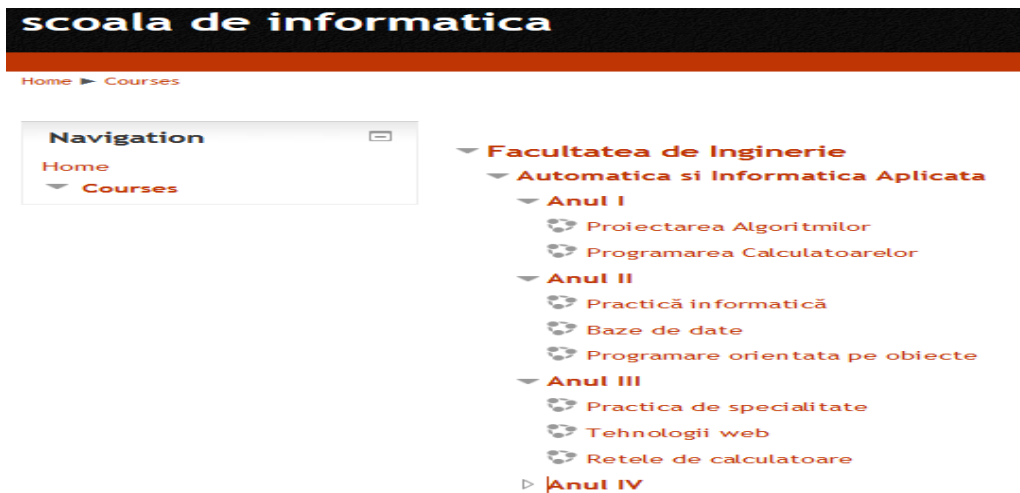


Figure 3. Programming courses page selection

In the first block of the course can enter general information about:

1. The course taught (is possible to build a web-page dedicated to this section)
2. Structure / Syllabus Course
3. The teacher who teaches the course
4. Resources in attending the course:
 - a. Documentation online about the contents provided in the course
 - b. Information about how to use course resources
 - c. Tutorials known online virtual environment (eg <http://www.w3schools.com/>, etc.)

d. Free video tutorials offered in the online

5. General Forum (for asynchronous communication with all participants in the course), what should go in the general listings and news about ongoing

6. Forum only students (ie a place where students can talk to each other), a place where students can talk about the information presented in the course and any information relating to the course.

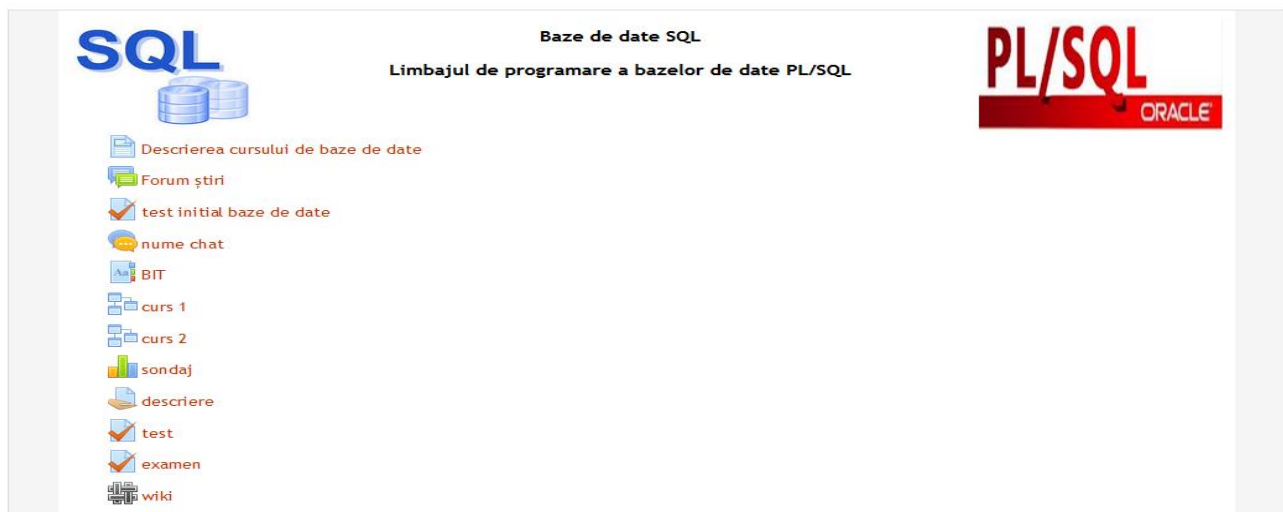
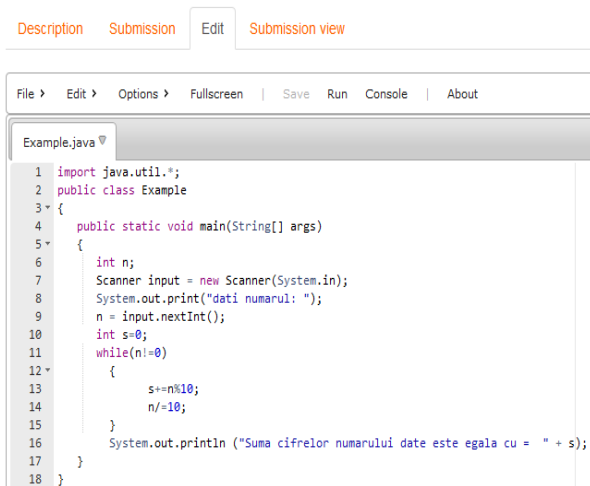


Figure 4 SQL and PL/SQL course main page

We installed a plugin [4] which can be used by the students for practical laboratory activities. They could compile and execute online any source code written by each student. Results are displayed on a console,

and files that have been executed and tested can be loaded and sent to the teacher for evaluation. This student has a very useful tool to check (self-evaluate) algorithms which she conceives. (Fig. 5 and Fig. 6)

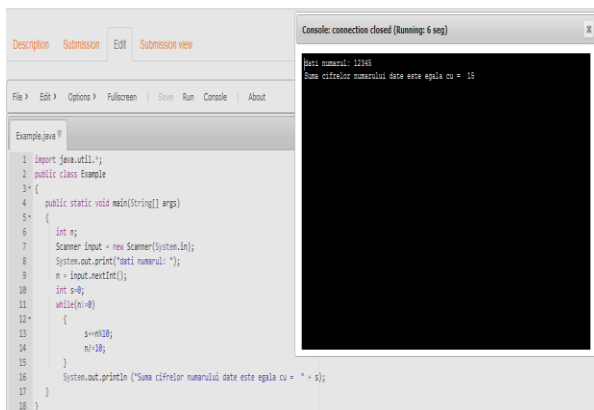


```

1 import java.util.*;
2 public class Example
3 {
4     public static void main(String[] args)
5     {
6         int n;
7         Scanner input = new Scanner(System.in);
8         System.out.print("dati numarul: ");
9         n = input.nextInt();
10        int s=0;
11        while(n!=0)
12        {
13            s+=n%10;
14            n/=10;
15        }
16        System.out.println ("Suma cifrelor numarului date este egala cu = " + s);
17    }
18 }

```

Figure 5 Example of Java source code



```

Console: connection closed (Running: 6 seg)
dati numarul: 11965
Suma cifrelor numarului date este egala cu = 26

```

Figure 6 Console for a Java Run program

In Moodle, all the available information within a course (for a specific subject) is organized in separate blocks. The first block is always devoted to general and administrative information, as well as support contents about the course: timetables, teachers, news, forums, chats and so on. The remaining blocks can be organized in different ways. In our course, each block corresponds to a specific area of the subject.

We have organized it as follows:

- First block: teachers, timetables, general forum, chats (one for each teacher), exams, lab description, course planning, homework (Fig. 4).
- Second block: General information's about databases, SQL types and variables
- Third block: Requests SELECT on a table, Functions related to one record, Functions related to multiple records
- Fourth bloc: Subqueries, Data Manipulation Language, Constraints, views

- Fifth block: General concepts in PL / SQL, Control the execution of a PL / SQL, Data Types in PL / SQL

- Sixth block: Cursors management in PL / SQL, Subprograms in PL / SQL, Packages in PL / SQL

- Seventh block: Using dynamic SQL commands

- Eight block: Triggers in PL / SQL, Error handling

This syllabus is the same as the one distributed the first day of class, so there is no difference between the CMS-oriented syllabus and the one used during the face-to-face course.

5. CONCLUSION

We believe Moodle is a great way for teachers to organize, manage and deliver course materials. From the didactic point of view, the usage of multimedia tools to create attractive activities makes the learning process friendlier for students. As a consequence, these activities increase the interest of the students in the study of Computer Programming. Teachers can provide students with a great amount of resources that usually they cannot show in the course due to the lack of time.

REFERENCES

- [1] Juan C. Rodríguez-del-Pino, Enrique Rubio-Royo, Zenón J. Hernández-Figueroa, A Virtual Programming Lab for Moodle with automatic assessment, Proceedings of The 2012 World Congress in Computer Science, Computer Engineering, and Applied Computing, 2012
- [2] Rodríguez-del-Pino, J.C.; Rubio-Royo, E.; Hernández-Figueroa, Z. Uses of VPL. 5th International Technology, Education and Development Conference (INTED), 2011, pp 743-748.
- [3] https://moodle.org/plugins/view.php?plugin=mod_vpl
- [4] https://moodle.org/plugins/view.php?plugin=mod_vpl